TITLE: SYSTEM AND METHOD ENABLING MULTIPLE
           PROCESSES TO EFFICIENTLY LOG EVENTS

INVENTOR: Panagiotis Kougiouris and Mac Vu

APPENDIX B

```
//'vclogserver.idl : IDL source for vclogserver.dll
//

// This file will be processed by the MIDL tool to
// produce the type library (vclogserver.tlb) and marshalling code.

import "oaidl.idl";
import "ocidl.idl";

    [
    object,
    uuid(EB01247E-A4DB-11D1-BFDE-00201829472A),
    helpstring("ICLogServer Interface"),
    pointer_default(unique)
    ]
    interface IHSLogServer : IUnknown
    {
        typedef struct tagLogInfo {
            long   m_cbSize;
            long   m_nProcessId;
            long   m_nThreadId;
            short  m_nLogLevel;
            short  m_nLogMask;
            long   m_microseconds;
            BSTR   m_bstrSource;
            DATE   m_dateTimestamp;
            LONGLONG m_performance;
        } HSLogInfo ;


        [id(4), helpstring("method Log")] HRESULT Log([in] HSLogInfo* logInfo, [in]BSTR LogS
tring);

        [id(5), helpstring("method Log")] HRESULT ClearLog();
    };

    [

        object,
        uuid(AC87A4FB-EA9B-11d1-8016-00201829472A),
        dual,
        oleautomation,
        helpstring("IHSLogServerProps Interface"),
        pointer_default(unique)
    ]
    interface IHSLogServerProps : IDispatch
    {
        [propget, id(1), helpstring("property FileName")] HRESULT FileName([out, retval] BST
R *pVal);
        [propput, id(1), helpstring("property FileName")] HRESULT FileName([in] BSTR newVal)
;
        [propget, id(2), helpstring("property LogLevel")] HRESULT LogLevel([out, retval] sho
rt *pVal);
        [propput, id(2), helpstring("property LogLevel")] HRESULT LogLevel([in] short newVal
);
        [propget, id(3), helpstring("property LogMask")]  HRESULT LogMask([out, retval] shor
t *pVal);
        [propput, id(3), helpstring("property LogMask")]  HRESULT LogMask([in] short newVal)
;
        [propget, id(4), helpstring("property MaxEventsCount")] HRESULT MaxEventsCount([out,
 retval] long *pVal);
```

1

```
        [propput, id(4), helpstring("property MaxEventsCount")] HRESULT MaxEventsCount([in]
long newVal);
    }


    [
        object,
        uuid(AC87A4FD-EA9B-11d1-8016-00201829472A),
        dual,
        oleautomation,
        helpstring("IHSLogServerConfig Interface"),
        pointer_default(unique)
    ]
    interface IHSLogServerConfig : IDispatch
    {
        [propget, id(340), helpstring("property Configuration")] HRESULT Configuration([out,
    retval] IHSLogServerProps** pVal);
        [propput, id(340), helpstring("property Configuration")] HRESULT Configuration([in]
IHSLogServerProps* newVal);
    }


    [
    object,
    uuid(C6771826-A4DE-11d1-BFDE-00201829472A),
    helpstring(" Interface"),
    pointer_default(unique)
    ]
    interface IHSLogServerNotify : IUnknown
    {
        import "oaidl.idl";
        HRESULT OnLogLevelChanged([in] short nNewLevel);
        HRESULT OnFileNameChanged([in] BSTR bstrFileName);
        HRESULT OnMaskChanged([in] short nNewMask);
    };
    [
    uuid(EB012471-A4DB-11D1-BFDE-00201829472A),
    version(1.0),
    helpstring("Healtheon Logging Server (HSSLOG) 1.0")
    ]
    library HSSLOG
    {
    importlib("stdole32.tlb");
    importlib("stdole2.tlb");

    interface IHSLogServerProps;


    [
    uuid(C6771827-A4DE-11d1-BFDE-00201829472A),
    version(1.0),
    hidden,
    nonextensible
    ]
    dispinterface  DIHSLogServerNotify {
    properties:
    methods:
        [id(0x00000001)]
        void OnLogLevelChanged([in] short intNewLevel);
        [id(0x00000002)]
        void OnFileNameChanged([in] BSTR bstrFileName);
```

```
    [id(0x00000003)]
    void OnMaskChanged([in] short nNewMask);
};


[
    uuid(EB012482-A4DB-11D1-BFDE-00201829472A),
    helpstring("CLogServer Class")
]
coclass Server
{
    [default] interface IHSLogServer;
    interface IHSLogServerConfig;
};

// This is non creatable,  used only so that VB
// can see the events
[
    uuid(354DDB60-66E6-11d2-8E37-00104B79DD7C),
    helpstring("CLogServer Class")
]
coclass Props
{
    [default] interface IHSLogServerProps;
    [source]  interface IHSLogServerNotify;
    [default, source] dispinterface DIHSLogServerNotify;
}

};
```